UNITED STATES DISTRICT COURT
WESTERN DISTRICT OF NEW YORK

MOOG INC.,

Plaintiff,

v.

SKYRYSE, INC., ROBERT ALIN PILKINGTON, MISOOK KIM, AND DOES NOS. 1-50,

Defendants.

CASE NO. 22-CV-00187

DECLARATION OF C. DOUGLASS LOCKE, PH.D.

TABLE OF CONTENTS

I, C. Douglass Locke, Ph.D., declare as follows:

I. INTRODUCTION

1. I have been retained as a technical expert by Latham & Watkins LLP, counsel for Skyryse, Inc. (Skyryse) to analyze and opine on the allegations of misappropriation, copying, or otherwise using trade secrets of Moog Inc. (Moog). I provide this declaration in support of Skyryse's Opposition to Moog's Motion to Compel Written Discovery Responses and Production of Documents. In particular, in this declaration I consider the statements made regarding these allegations by Mr. Kevin M. Crozier.

2. I am over the age of 18 years. I have personal knowledge of the matters set forth herein and if called as a witness, I could and would competently testify as to all facts set forth herein.

3. In reaching the opinions stated herein, I have considered documents and filings made available in this case in the context of my own education, training, research, knowledge, as well as my personal and professional experience.

II. BACKGROUND AND QUALIFICATIONS

4. My background and qualifications are stated more fully in my curriculum vitae. *See* Ex. 1. As reflected in my curriculum vitae (and as explained below), I have significant professional experience with software and software development for aviation computation, including safety-critical software.

5. I am a full-time consultant with a variety of clients, such as the United States Office of the Secretary of Defense, the Jet Propulsion Laboratory, Raytheon Corporation, and Baxter International, Inc. My consulting work involves a wide variety of systems, software, and the hardware used in computer systems, especially embedded computer systems, including those used in avionics.

6. Embedded computers consist of processors incorporated within other products such as aircraft, spacecraft, ground vehicles, maritime platforms, handheld devices, and medical devices. My consulting work therefore includes extensive experience with such diverse technologies as avionics, aircraft/spacecraft sensors, communication (both wired and wireless), and actuators, as well as more common software and systems technologies including displays, cameras, operator interfaces, databases, security, data structures, computer and network architectures, and software engineering.

7. I have designed, specified, and consulted on aircraft and rotorcraft systems over a period of more than 40 years, including work on such systems as unmanned spacecraft, manned spacecraft, Navy helicopters, spacecraft ground control, air traffic management.

8. For example, I was the software architect for the U.S. Navy's SH-60B Light Airborne Multipurpose Systems (LAMPS) helicopters. This project included work on Kaman SH-2 Seasprite and Sikorsky SH-3 helicopters in its early stages.

In this role, I had technical responsibility for all of the software controlling sensors, hardware interfaces, human interfaces, communications interfaces, and data storage devices on the platform.

9. Beyond my experience in aircraft and spacecraft, I have an extensive background in a wide variety of computing and data management platforms, including areas such as systems architecture, design, implementation, and deployment of software, system performance, related international standards, safety-critical standards, computer language operations (e.g., C, C++, Java, Python, HDML, Ada, and JavaScript), software engineering maturity, and software organization.

III. Evidence Regarding Crozier's Allegations of Copying Moog Documents

10. Mr. Crozier states in his declaration that he has been "granted access to certain electronic devices and data turned over to iDS by all parties in the case through inspection laptops and iDS' remote virtual machine software." (ECF 210-1, ("Cozier Decl."), ¶ 10) He further states that his opinions of alleged copying were based on that access. I have also been granted access to these same electronic devices and data, and my conclusions are based on that access as well as on statements made by Mr. Crozier.

11. In his declaration (Cozier Decl. ¶ 11), Mr. Crozier identifies two alleged instances of "direct copying of Moog software and documents."

12. The first of these two alleged instances consist of directories and files from two Windows laptops containing software components of a desktop test environment, one purported to have been used by Mr. Pilkington during his time at Moog and the other from his time at Skyryse.

13. The second of these two alleged instances consist of directories and files regarding software process checklist templates from two Macintosh laptops, one purported to have been used by Mr. Pilkington during his time at Moog and the other from his time at Skyryse.

14. I consider these two issues in turn below for the purposes of responding to Mr. Crozier's declaration submitted in support of Moog's motion to compel. I reserve the right to further analyze or opine on these issues and to revise my opinions based on the review of further information.

A. Moog's Version of a Desktop Test Environment

15. The files identified by Mr. Crozier regarding a desktop test environment are found in three directories on the two Windows laptops from Moog and Skyryse.

16. Mr. Crozier states in his declaration that he has "noted several examples of Skyryse's direct copying of Moog software and documents." However, the information reviewed by Mr. Crozier is insufficient to conclude that either direct copying occurred or that the software and documents were, in fact,

Moog's trade secrets or proprietary to Moog. The files identified from the Moog and Skyryse computers by Mr. Crozier do show signs of having been derived from a common source. However, without further information, I disagree that Mr. Crozier has identified evidence of "direct copying" of the Moog files from the corresponding Skyryse files. While similarities between computer files might indicate copying from one to the other, they also may indicate the existence of a third source from which both files have been derived, and Mr. Crozier fails to identify any evidence to conclude with certainty that these files are direct copies of one another.

17. In addition, Mr. Crozier, when discussing the DO-178C standard used to certify safety-critical software by the Federal Aviation Administration and which specifies how such software should be created, tested, and verified, appears to imply in his declaration that the programs in these directories are specifically designed to implement the DO-178C testing requirements. (*See* Crozier Decl. ¶ 12.) I disagree. While the software tools in these directories are involved in automated testing that can be used to satisfy the testing requirements of DO-178C, these simple programs follow well-known automated test paradigms. These same test paradigms are used in a wide variety of embedded and other software far beyond DO-178C. Nothing in these programs is necessarily specific to developing flight control software.

18. In addition, while I have not identified evidence to confirm with certainty that the files identified by Mr. Crozier are direct copies of each other, evidence in the files I reviewed suggests the opposite, and that some of these files—including the larger source code files—were derived from a common source that predates either Mr. Pilkington or Ms. Kim's employment at Moog, and also may predate Moog having first obtained these files.

19. For example, Mr. Crozier identifies six files in the directory [redacted] Every program in this directory, all of which are written in the C language, contains a header comment block, similar to most C programs, that includes a Revision History section. For each of these C programs, this Revision History includes its initial creation date and indicates the author of each file as Alin Pilkington. Below is a table showing the initial creation date and author for each of these six files from its header comment block. I note that these creation dates predate Mr. Pilkington or Ms. Kim's employment by Moog, which I understand to have begun in 2012 and 2013, respectively, rendering it unclear at best where these files were created or whether they were created at Moog. Rather, the information suggests that Mr. Pilkington may have created these files prior to his time at Moog.

File	Initial Creation Date	Author
[redacted]	February 6, 1997	Alin Pilkington
[redacted]	September 26, 1996	Alin Pilkington
[redacted]	September 26, 1996	Alin Pilkington
[redacted]	September 26, 1996	Alin Pilkington
[redacted]	November 9, 2001	Alin Pilkington
[redacted]	September 26, 1996	Alin Pilkington

20. Mr. Crozier also identifies ten files in the directory: [redacted] Again, I note that each program in this directory, written in the C language, contains a header comment block, similar to most C programs, that includes a Revision History section, including their initial creation date. Below is a table showing the ten files their indicated initial creation dates, and their indicated author. All but two of these creation dates predate Mr. Pilkington or Ms. Kim's employment by Moog as alleged by Moog in the Complaint (¶¶ 12-13). The information again suggests that Mr. Pilkington may have created these files prior to his time at Moog, which also may predate Moog having first obtained these files.

File	Initial Creation Date	Author
[redacted]	August 9, 2013	Alin Pilkington
[redacted]	August 9, 2013	Alin Pilkington
[redacted]	November 9, 2001	Alin Pilkington
[redacted]	September 26, 1996	Alin Pilkington
[redacted]	No date available	Alin Pilkington
[redacted]	April 6, 2002	Alin Pilkington
[redacted]	January 13, 1997	Alin Pilkington
[redacted]	January 13, 1997	Alin Pilkington
[redacted]	November 9, 2001	Alin Pilkington
[redacted]	September 12, 1996	Alin Pilkington

21. As previously mentioned, the files identified in this directory from the Moog and Skyryse computers by Mr. Crozier do show signs of having been derived from a common source. However, without further information, I disagree that Mr. Crozier has shown evidence of "direct copying" of the Skyryse files from the corresponding Moog files. Deriving file contents from a common source might indicate copying from one to the other, but might also indicate the existence of a third source from which both files have been derived.

22. In addition, Mr. Crozier identifies eight files in the directory: [redacted] It appears that many of the functions among these eight files are redundant, so even if they were copied from each other, this would achieve little if any gain in productivity – one developing these eight files from scratch today would probably not develop more than a few of them. Further, creating from scratch any of the files listed in any of the directories described above or otherwise referenced in Mr. Crozier's declaration would not require any technical knowledge involving avionics, flight control, aircraft systems, embedded software, safety-critical software, or the DO-178C standard under which an FAA-certifiable system is developed. Only a general knowledge of Visual Studio (VS) interfaces, C programming, and Software Engineering (the discipline that covers manual and automated test plans, specifications, procedures, and reporting) would be needed. As I noted previously, such knowledge could be acquired from a standard Software Engineering textbook, and is readily available to people in this industry and hardly a secret. A competent Software Engineer with knowledge at the level of a university undergraduate Computer Science degree and a suitable Software Engineering textbook[1] would

[1] Such as, Sommerville, *Software Engineering*, Addison Wesley, Ninth Edition, Addison-Wesley, 2011, or Cooling, *Software Engineering for Real-Time Systems*, Addison-Wesley, 2003.

have sufficient understanding of automated testing techniques to produce these files with little effort.

B. Moog's Software Process Checklist Templates

23. All of the files regarding software process checklist templates identified by Mr. Crozier are in the single directory "Software Checklists," whose full path name in the Moog system is [redacted] and in the Skyryse system is [redacted] All of these files are Excel spreadsheets. There are ten spreadsheets in this directory.

24. These spreadsheet files identified by Mr. Crozier do nothing more than list a set of criteria to be used by a review team to review various artifacts at different phases of software development. None of these spreadsheets appear to contain any Moog or Skyryse data, and the checklist criteria contained in them represent ordinary Software Engineering textbook criteria that are not specific to any particular project, including to DO-178C. As such they have minimal value and could be readily created from publicly available information.

25. For example, the file [redacted] intended to guide a review team for a source code review, contains the statement, [redacted]

[redacted] This statement is a general guide to assessing the correctness of the source code, but, like the other spreadsheets, contains no Moog data or Skyryse proprietary data that I am aware of, and the checklist criteria represent ordinary Software Engineering textbook criteria[2] and are not specific to any particular project nor to DO-178C. As such these criteria could be readily created from publicly available information.

IV. CONCLUSION

26. I reserve the right to respond to any declarations that are submitted by Plaintiff's expert witness(es), to any testimony by Plaintiff's fact or expert witnesses or to any further decisions, including amendments, from the Court.

[2] These, and many other source code quality criteria can be found, for example, in Sommerville (op. cit.), Cooling (op. cit), van Vliet, *Software Engineering*, Third Edition, Wiley, 2008, Siewert, *Real-Time Embedded Systems and Components*, Thomson, 2007, and Edwards, *Real-Time Structured Methods – Structured Analysis*, Wiley, 1993

I declare under penalty of perjury that the foregoing is true and correct. Executed in Denver, NC on August 17, 2022.

C. Douglass Locke, Ph.D.

Exhibit 1

Curriculum Vitae

C. Douglass Locke, Ph.D.
Professor, and Software and Systems Consultant

Locke Consulting LLC.
531 Brentwood Rd, #114
Denver, NC 28037-0269
704-746-3647 Office

704-301-7074 Cell
704-253-4957 Fax
doug@douglocke.com
www.douglocke.com

PROFESSIONAL EXPERIENCE

Since 2004, Dr. Locke, President of Locke Consulting LLC., has been a full-time consultant with a variety of clients involved primarily in Military, Aerospace, Industrial Control, and Telecommunications systems. His extensive technical and business background in areas such as software engineering, software performance, real-time architecture, design, implementation, and deployment, international standards, software engineering maturity, and software development management and organization has made a major impact on organizations creating complex systems. He has used and analyzed more than 25 computer languages, including Java, C, C++, Python, HTML, JavaScript, as well as a variety of computer assembly languages. His clients include over 40 companies and agencies, including the Naval Systems Warfare Center, DoD Office of the Secretary of Defense, Raytheon Corporation, Jet Propulsion Laboratory, Baxter International, Motorola (now Google), and Sun Microsystems (now Oracle).

From 2016 to 2021, he served as a Visiting Professor at Davidson College, Davidson, North Carolina in the Department of Mathematics and Computer Science. In this capacity, he developed and/or taught courses including Programming and Problem Solving, Software Engineering, Data Structures, and Computer Architecture & Organization. Previously he also served as an Adjunct Professor at Lenoir-Rhine University in Hickory, North Carolina where, in the department of Mathematics and Computer Science, he taught a class on C++.

In addition to his work with clients involved in specifying, building, and deploying software-intensive systems, including embedded and real-time systems, he also has experience serving as an Expert (including expert reports, depositions and courtroom testimony) in intellectual property-related litigation including patents, copyrights, trade secrets, and contracts. Recent cases involve patents for computer controls for heavy-duty truck transmissions, patents for safety/control systems for hospital proton beam therapy systems, smartphones, and trade secrets associated with vehicle tracking devices.

Prior to February 2004, Doug served as the Vice President of Technology for TimeSys Corporation, a producer of Embedded Linux distributions and software development tools. In this position, he had responsibility for defining the technology base for the corporation and extending it to meet customer requirements. While at TimeSys, he also served as a key consultant and instructor for TimeSys critical projects and classes,

provided key customers with systems and software consultation, and provided an external presence for the corporation in the embedded systems community with frequent publications and public presentations.

Prior to January 2000, Dr. Locke served as the Chief Scientist of the Lockheed Martin Systems Solutions organization. His responsibilities there were to define the software and systems technology focus for a new business thrust involving efforts by major industries worldwide transitioning from mainframe-based, "stovepipe" application architectures to distributed architectures. Most of the customers for this formerly exclusively aerospace organization required exploitation of the same kind of real-time, limited footprint, embedded systems experience for which Lockheed Martin is well-known. In addition to his work with major commercial customers, he also consulted extensively with other major Lockheed Martin projects across the corporation.

From September, 1989 to August, 1998, he was the Chief Scientist of IBM's (and Lockheed Martin's after IBM's FSD was acquired by Lockheed Martin) Software & Systems Resource Center. His responsibilities there centered primarily on management, systems, and software consulting across IBM and Lockheed Martin, guiding major projects' management as well as the technology dissemination activities at the SSRC, reviewing / updating the SSRC's educational offerings, and leading / participating in major program reviews / audits. A wide variety of projects were supported such as the Space-Based Infrared System (SBIRS), Sustaining Base Information System (SBIS), Air Traffic Management, GPS, various avionics systems, Close Combat Tactical Trainer (CCTT), Light Airborne Multipurpose System (LAMPS), and *Airborne Warning and Control System (AWACS)*.

At IBM and Lockheed Martin, Dr. Locke led and participated in many management and technical review teams evaluating major government projects with respect to their ability to successfully manage and complete complex system development and deployment. At TimeSys Corporation, where Dr. Locke served as VP Technology and VP Engineering, he had an active role in all aspects of the corporate leadership, including CEO recruitment, venture capital interfaces, marketing, sales, technology, engineering, and customer satisfaction.

From 1981 through 1999, Dr. Locke acted as a consultant to all levels of management, senior technical staff, and customers for a wide variety of Lockheed Martin, Loral, IBM, TimeSys, and other system development, research, product development, and standards activities. This consultation covered both management and technical issues related to real-time software development, open systems, object-oriented systems, Ada, C, C++, and Java applications, and general systems performance. Projects on which he has spent substantial amounts of time include the Air Force's Theater Battle Management Core System, the Global Positioning System, Air Traffic Control, the UK Ministry of Defense Merlin System (an Anti-Submarine Warfare helicopter), the Navy's LAMPS ASW helicopter system, the Army's Rotorcraft Pilot's Associate, the Army's Sustaining Base Information System, and the Joint Tactical Radio System.

Throughout his consulting work, he frequently led and participated in major project audits and reviews, especially for major proposals, projects experiencing cost/schedule perturbations, and projects experiencing performance difficulties.

In addition to his other ongoing responsibilities, Dr. Locke has led and/or participated in a number of international standards including the Portable Operating System Interface (POSIX) standard (IEC/ISO 9945, the Real-Time Common Object Request Broker Architecture (CORBA) by the Object Management Group, the Real-Time Specification for Java (RTSJ JSR-1) under the Java Community Process, and the Safety Critical Java (SCJ JSR-302) also under the Java Community Process.

Between 1992 and 1995, he served as a regular instructor for the U.S. Air Force's *Operation Bold-Stroke*, teaching software procurement and management principles to senior officers (Generals, Colonels, and equivalent civilian staff only.)

EDUCATION

Ph.D., Computer Science, Carnegie-Mellon University, 1986. Dissertation Title: B*est-Effort Decision Making for Real-Time Scheduling*

B.A., Physics, Kalamazoo College, 1965

PUBLICATIONS

- *JSR-000302 Safety Critical Java™ Specification*, **EDR-3, Java Community Process**, January, 2017
- *Java for Safety-Critical Applications*, **Proceedings of SafeCert 2009**, York, UK, 2009
- *Java Technology Comes to Real-Time Applications*, **Proceedings of the IEEE**, July, 2003
- *Real-Time Database Applications and System Characteristics, in* ***Real-Time Database Systems: Architecture and Techniques****, Kluwer Academic Publishers, 2001*
- *Software Architectures for Hard Real-Time Applications: Cyclic Executives vs. Fixed Priority Executives,* ***Real-Time Systems Journal****, April, 1992*
- *Building a Predictable Avionics Platform in Ada: A Case Study,* ***Proceedings of the IEEE Real-Time Systems Symposium****, December, 1991*
- *Scheduling in Real-Time,* ***UNIX Review****, pp. 48-54, Vol 8, No. 9, September, 1990*

- *Predictable Real-time Avionics Design Using Ada Tasks and Rendezvous, Fourth International Workshop on Real-Time Ada Issues, ACM-SIGAda* ***Ada Letters****, June, 1990*

- *Priority Inversion and its Control: An Experimental Investigation, Second International Workshop on Real-Time Ada Issues, ACM-SIGAda* ***Ada Letters****, June, 1988*

- *A Practical Application of the Priority Ceiling Protocol in a Real-Time System, Second International Workshop on Real-Time Ada Issues, ACM-SIGAda* ***Ada Letters****, June, 1988*

- *The Avionics Software Architecture Impact on System Architecture,* ***Proceedings of the AGARD*** *(Advisory Group for Aerospace Research & Development) Conference, Sponsored by the North Atlantic Treaty Organization, April, 1987*

- *Problems in Ada Runtime Task Scheduling, First International Workshop on Real-Time Ada Issues, ACM-SIGAda* ***Ada Letters****, May, 1987*

- *A Time-Driven Scheduling Model for Real-Time Operating Systems,* ***Proceedings of the Real-Time Systems Symposium****, December, 1985*

- *The Ada Programming Support Environment, in* ***Ada: A Tutorial****, Published by IEEE, 1983 (Previously published in IBM FSD's* ***Software Engineering Exchange****)*

- *Real-Time Java Moving Into the Mainstream,* **RTC Magazine**, available at http://www.rtcmagazine.com/home/printthis.php?id=100069, 2006

- *Real-Time Architecture—Past, Present, and Future,* Mark Gerhardt and Doug Locke, **2005 Embedded Systems Conference** San Francisco, March 10, 2005

- *Architecture for Predictable Systems,* **2004 Embedded Systems Conference** San Francisco Paper, available at http://www.techonline,com/community/member_company/member/1034/content 40291, 2004

- *Introduction to Open Architecture for Real-Time Systems,* **The Open Group**, available at www.opengroup.org, July 2003

- *Priority Inheritance: The Real Story*, **Linux Devices**, available at http://www.linuxdevices.com/articles/AT5698775833.html, July 16, 2002

- *A TimeSys Response on Real-Time Linux*, **Linux Devices**, available at http://www.linuxdevices.com/articles/AT5625209055.html, May 20, 2002

- *What is real-time?*, **Linux Devices**, available at http://www.linuxdevices.comarticles/AT6090565653.html, Sept. 14, 2000

- *IEEE standard for information technology-portable system interface* (POSIX)-part I: *system application program interface (API)-amendment J: advanced realtime extensions [C language],* IEEE Std 1003.1j-2000, pp. 1-88, **IEEE**, 2000

- *IEEE standard for information technology-portable system interface* (POSIX)-part I: *system application program interface (API)-amendment J: advanced realtime extensions [C language],* IEEE Std 1003.1-2000, **IEEE**, 2000.

- *An Architectural Perspective of Real-Time Ada Applications*, Ada-Europe 1999: 1-11., **Springer-Verlag**, 1999

- *A Real-Time, Fault Tolerant CORBA Implementation: A Case Study* (Abstract), C. Douglass Locke and Thomas A. North, **Fourth International Workshop on Object-Oriented Real-Time Dependable Systems** (WORDS '99), 1999

- *Error Propagation Analysis of Real-Time Data Intensive Applications*, Tei-Wei Kuo, Doug Locke, Farn Wan, **IEEE Real Time Technology and Applications Symposium**: 166-171, 1997

- *Future Distributed Embedded and Real-Time Data Intensive Applications Will Be Adaptive: Meanings, Challenges and Research Paradigms (Panel),* Aloysius K. Mok, Constance L. Heitmeyer, Kevin Jeffay, Michael B Jones, C. Douglass Locke, Ragunathan Rajkumar, **International Conference on Distributed Computing Systems (ICDCS)**:182-184, 1995

- *Fault Tolerant Applications Systems: A Requirements Perspective, Hardware and Software Architectures for Fault Tolerance*, Fault Tolerant Workshop Proceedings, Le Mont Saint Michel, France, **Springer-Verlag**, Pages: 21 – 25, 1994